



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07160490 A**(43) Date of publication of application: **23.06.95**

(51) Int. Cl.

G06F 9/06
G06F 9/45
G06F 11/28

(21) Application number: **05302087**(22) Date of filing: **01.12.93**(71) Applicant: **NIPPONDENSO CO LTD**

(72) Inventor: **KAWAKAMI HIDEHIKO**
OKUMURA MITSUO
SUZUKI MASAHIRO

(54) **CODING ASSISTANCE DEVICE**

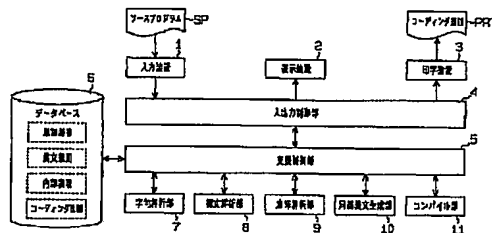
brought under the display control of a display device 2.

(57) Abstract:

COPYRIGHT: (C)1995,JPO

PURPOSE: To assist the description of a source program in an optional program language by visually displaying an inputted source program and the source program generated as synonymous different sentences by analyzing the inputted source program in the form of object codes obtained by compiling them under the same conditions, and selectively storing the object codes.

CONSTITUTION: In a data base 6, a word dictionary, a syntax rule, internal expressions, and a code rule as to the object language are stored, and a word and phrase analysis part 7, a syntax analysis 7-7-N part 8, and a meaning analysis part 9 take analyses on the basis of information in the data base 6. The process contents of the source program SP which are made clear are passed to a homonymous different sentence generation part 10 to generate the source program which has the same process contents with the source program SP and is different in describing method. A compilation part 11 compiles those programs individually to generate the object codes, which are converted into an assembler expansion form by an assistance control part 5 and



Ref.
 #4
 on JP
 OA.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-160490

(43) 公開日 平成7年(1995)6月23日

(51) Int.Cl.⁶

G 0 6 F 9/06

9/45

11/28

識別記号

5 3 0 P

C

E

庁内整理番号

9367-5B

9367-5B

9290-5B

9292-5B

F I

技術表示箇所

G 0 6 F 9/ 44

3 2 2 M

審査請求 未請求 請求項の数5 O L (全 14 頁)

(21) 出願番号

特願平5-302087

(22) 出願日

平成5年(1993)12月1日

(71) 出願人 000004260

日本電装株式会社

愛知県刈谷市昭和町1丁目1番地

(72) 発明者 川上 英彦

愛知県刈谷市昭和町1丁目1番地 日本電装株式会社内

(72) 発明者 奥村 光雄

愛知県刈谷市昭和町1丁目1番地 日本電装株式会社内

(72) 発明者 鈴木 巨裕

愛知県刈谷市昭和町1丁目1番地 日本電装株式会社内

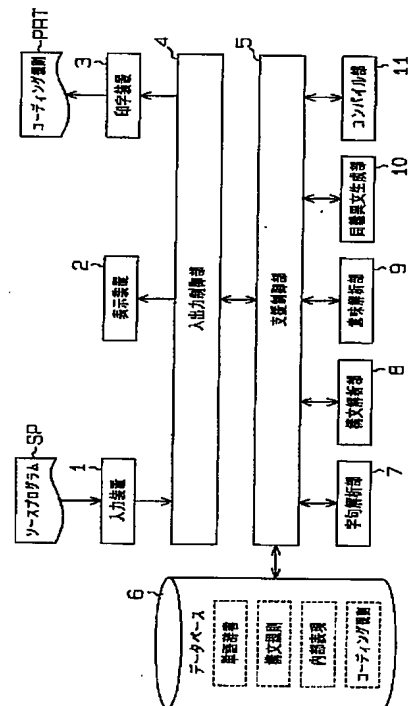
(74) 代理人 弁理士 恩田 博宣

(54) 【発明の名称】 コーディング支援装置

(57) 【要約】

【目的】 任意のプログラム言語によるソースプログラムの記述を支援し、しかもより効率のよいオブジェクトコードの生成を可能ならしめるコーディング支援装置を提供する。

【構成】 入力されたソースプログラムの字句、構文、及び意味の解析を通じて明らかにされたソースプログラムの制御構造をもとに同義異文を生成する。この生成した同義異文は入力ソースプログラムと共に同一条件にてコンパイルするとともに、それら生成された各オブジェクトコードについてはこれをニモニック表示してユーザの選択を求める。そして、ユーザによる選択のあったオブジェクトコードに対応するソースプログラムの記述手法をコーディング規則としてデータベースに記憶し、出力操作があった場合には、この記憶したコーディング規則を一覧表として印字若しくは表示する。



【特許請求の範囲】

【請求項1】 任意の手続が記述されたソースプログラムが入力される入力手段と、

この入力されたソースプログラムの字句を解析して同ソースプログラムを当該記述言語としての一連のトークンに分解する字句解析手段と、

この分解されたトークンに基づき当該ソースプログラムの構文を解析する構文解析手段と、

この解析された構文の論理構造や文の内容に基づき当該ソースプログラムに記述された手続の意味を解析する意味解析手段と、

これら各解析結果に基づき、当該記述言語において当該ソースプログラムに記述された手続と意味的に同一の手続からなる異なるソースプログラムを生成する同義異文生成手段と、

前記入力されたソースプログラム及び前記同義異文として生成されたソースプログラムを同一条件にてコンパイルするコンパイル手段と、

これらコンパイルによって生成されたオブジェクトコードを、認識し得る所定の表現にて可視表示する表示制御手段と、

これら表示されたオブジェクトコードに対する適宜の選択操作に基づき、その選択されたオブジェクトコードに対応するソースプログラムの記述手法を所望のコーディング規則としてこれを適宜の記憶手段に格納するコーディング規則格納手段と、

を具えることを特徴とするコーディング支援装置。

【請求項2】 請求項1に記載のコーディング支援装置において、

適宜の出力操作に基づき、前記記憶手段に記憶されたコーディング規則についての情報を所定の形式にて出力する出力制御手段、

を更に具えることを特徴とするコーディング支援装置。

【請求項3】 任意の手続が記述されたソースプログラムが入力される入力手段と、

この入力されたソースプログラムの字句を解析して同ソースプログラムを当該記述言語としての一連のトークンに分解する字句解析手段と、

この分解されたトークンに基づき当該ソースプログラムの構文を解析する構文解析手段と、

この解析された構文の論理構造や文の内容に基づき当該ソースプログラムに記述された手続の意味を解析する意味解析手段と、

これら各解析結果に基づき、当該記述言語において当該ソースプログラムに記述された手続と意味的に同一の手続からなる異なるソースプログラムを生成する同義異文生成手段と、

前記入力されたソースプログラム及び前記同義異文として生成されたソースプログラムを同一条件にてコンパイルするコンパイル手段と、

これらコンパイルによって生成されたオブジェクトコードのコードサイズを比較する比較手段と、

この比較の結果、よりコードサイズの小さいオブジェクトコードに対応するソースプログラムの記述手法を所望のコーディング規則としてこれを適宜の記憶手段に格納するコーディング規則格納手段と、

を具えることを特徴とするコーディング支援装置。

【請求項4】 請求項3に記載のコーディング支援装置において、

10 適宜の出力操作に基づき、前記記憶手段に記憶されたコーディング規則についての情報を所定の形式にて出力する出力制御手段、

を更に具えることを特徴とするコーディング支援装置。

【請求項5】 任意の手続が記述されたソースプログラムが入力される入力手段と、

この入力されたソースプログラムの字句を解析して同ソースプログラムを当該記述言語としての一連のトークンに分解する字句解析手段と、

20 この分解されたトークンに基づき当該ソースプログラムの構文を解析する構文解析手段と、

この解析された構文の論理構造や文の内容に基づき当該ソースプログラムに記述された手続の意味を解析する意味解析手段と、

これら各解析結果に基づき、当該記述言語において当該ソースプログラムに記述された手続と意味的に同一の手続からなる異なるソースプログラムを生成する同義異文生成手段と、

30 適宜の出力操作に基づき、この同義異文として生成されたソースプログラムを所定の形式にて出力する出力手段と、

を具えることを特徴とするコーディング支援装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、任意のプログラム言語によるソースプログラムの記述を支援するコーディング支援装置に関する。

【0002】

40 【従来の技術】 プログラム言語とは周知のように、コンピュータによる各種の処理手続を記述するための言語である。また、それらプログラム言語として、C（シー）言語をはじめ、C++（シー・プラス・プラス）、Pascal（パスカル）、COBOL（コボル）、アセンブラ等々、いわゆる高級言語と称されるものから低級言語と称されるものまで、数多くの言語があることもよく知られている。

50 【0003】 ところで、これらのプログラム言語には通常、同一の処理手続を定義するための複数の記述方法が存在する。そして、それら記述方法の何れを採用するかは、その言語ユーザであるプログラマに任されているのが普通である。

【0004】一方、ある任意のプログラム言語において、たとえ同一の処理手続として記述される場合であれ、その記述方法が異なれば、その後のコンパイルによって得られるオブジェクトコードの内容やコードサイズも微妙に異なったものとなる。

【0005】このように、プログラム言語は一般に、ある処理手続を記述するのに如何なる記述方法を採用するかはプログラマの意思に任されているとはいえ、その記述方法によっては、可読性はもとより、オブジェクトコードのコードサイズ、或いはその実行速度等に違いが来すようになる。

【0006】勿論、熟練したプログラマであれば、コンパイルによって生成されるオブジェクトコードを予測しつつ、そのコードサイズ等をより効率的なものとし得る記述方法をその記述しようとする処理内容に応じてその都度経験的に選択することはできる。

【0007】しかし、経験の少ないプログラマにとっては、生成されたオブジェクトコードを見てみなければそうした判断を下せないのが実情である。このように、生成されたオブジェクトコードを確認しつつ、それらソースプログラムの記述を書き代えていたのでは、プログラムの生産能率も低下せざるを得ない。

【0008】また、一連の関連したプログラムを複数のプログラマが協同して作成するような場合には、それら各プログラマによる記述方法の違いに基づく可読性の悪化やオブジェクトコードのばらつき、更にはそれら記述方法の修正によって引き起こされる生産能率の低下等もより一層深刻なものとなる。

【0009】そこで従来は、プログラム言語の記述方法、すなわちコーディング方法にある一定の規則を設けることで、こうした不都合に対処するようにしている。例えば特開平3-5837号公報に記載の装置では、こうしたコーディング規則をファイルとして予め記憶手段に登録しておき、記述されたソースプログラムがこのコーディング規則に合致しているか否かを自動的に検査するようにしている。

【0010】

【発明が解決しようとする課題】このように、予め登録したコーディング規則に基づいてソースプログラムの記述を検査するようにすれば、少なくともこれを利用するユーザ（プログラマ）にとって、それら複数のユーザ間で統一され且つ可読性の高いソースプログラムを得ることは確かにできる。

【0011】しかし、上記従来の装置では、例えばコードサイズの小さいより効率的なオブジェクトコードを得るために、ソースプログラムとしてどのような記述をすればよいか等、より望ましいコーディング規則を得るための手法については何等言及されていない。

【0012】したがって、上記登録されたコーディング規則を修正するにしろ、どのような基準に基づき、どの

ような修正を行えばよいか、等々については不明であり、結局は、熟練したプログラマの経験と勘に頼らざるを得なかった。

【0013】この発明は、こうした実情に鑑みてなされたものであり、任意のプログラム言語によるソースプログラムの記述を汎用的に支援し、しかもより効率のよいオブジェクトコードの生成を可能ならしめるコーディング支援装置を提供することを目的とする。

【0014】

10 【課題を解決するための手段】こうした目的を達成するため、この発明では、(a)任意の手続が記述されたソースプログラムが入力される入力手段と、(b)この入力されたソースプログラムの字句を解析して同ソースプログラムを当該記述言語としての一連のトークンに分解する字句解析手段と、(c)この分解されたトークンに基づき当該ソースプログラムの構文を解析する構文解析手段と、(d)この解析された構文の論理構造や文の内容に基づき当該ソースプログラムに記述された手続の意味を解析する意味解析手段と、(e)これら各解析結果に基づき、当該記述言語において当該ソースプログラムに記述された手続と意味的に同一の手続からなる異なるソースプログラムを生成する同義異文生成手段と、(f)前記入力されたソースプログラム及び前記同義異文として生成されたソースプログラムを同一条件にてコンパイルするコンパイル手段と、(g)これらコンパイルによって生成されたオブジェクトコードを、認識し得る所定の表現にて可視表示する表示制御手段と、(h)これら表示されたオブジェクトコードに対する適宜の選択操作に基づき、その選択されたオブジェクトコードに対応するソースプログラムの記述手法を所望のコーディング規則としてこれを適宜の記憶手段に格納するコーディング規則格納手段と、を具えてコーディング支援装置を構成する。

【0015】

40 【作用】ソースプログラムとして入力されたユーザ（プログラマ）による記述は、上記字句解析手段、構文解析手段、及び意味解析手段を通じて、トークン（綴り）と共に、その構文並びに意味が解析され、その所定の制御構造が明らかにされる。上記同義異文生成手段は、こうして明らかにされた当該ソースプログラムの制御構造をもとに、意味的に同一の手続からなる異なるソースプログラム、すなわち同義異文を生成する部分である。例えば、明らかにされた当該ソースプログラムの制御構造がループ制御として定義されたものであれば、この同義異文生成手段では、同じループ制御を実現する他の構文構造のテンプレート（ひな型）をデータベース等から検索し、同明らかにされた当該ソースプログラムの具体的な処理内容や値をこのテンプレートに埋め込むことで同義異文を自動生成する。

50 【0016】上記構成によればこのように、ユーザによって記述されたソースプログラムの他に、該プログラム

と実質的に同一の手続が記述されたいわば他の候補としてのソースプログラムが同義異文として自動生成される。

【0017】一方、これら候補としての両プログラムは、上記コンパイル手段を通じて同一条件にてコンパイルされ且つ、上記表示制御手段を通じてそのコンパイル結果が例えばニーモニック表現にて可視表示される。これにより、オブジェクトコードとして見たそれら両ソースプログラムの違い、優劣等は瞭然となる。

【0018】したがってその後、これらオブジェクトコードについての選択操作が行われ、上記コーディング規則格納手段を通じて記憶手段に格納されるソースプログラムの記述手法とは自ずと、ユーザが所望した手続についてのより好ましいコーディング規則を示すものとなる。ここに、熟練したプログラマによる経験や勘などは一切必要とされない。

【0019】なお、こうして格納されたコーディング規則の用途は任意であり、例えばこれを従来の装置において記憶手段に登録するコーディング規則ファイルとして用いたり、或いは印字したり、表示装置に表示させるなどして、いわばガイド的に用いることもできる。

【0020】特に、上記の構成に加えて、(i)適宜の出力操作に基づき、前記記憶手段に記憶されたコーディング規則についての情報を所定の形式にて出力する出力制御手段、を更に具え、上記生成された各種手続別の好ましいコーディング規則を、例えば表として一覧するものとすれば、各ユーザ（プログラマ）共、以後、この出力されたコーディング規則をもとにソースプログラムを記述するようにすることで、前述した可読性の悪化やオブジェクトコードのばらつき、更にはそれら記述方法の修正等によって引き起こされる生産能率の低下などといった問題も良好に解消されるようになる。

【0021】また、上記の構成では、比較表示されるオブジェクトコードに基づき、ユーザがその選択を行うものとしたが、オブジェクトコードのコードサイズであれば、上記コンパイルの結果に基づき、同支援装置自身で、その大小を認識することができる。すなわちこの場合、上記(g)の表示制御手段、及び(h)のコーディング規則格納手段に代えて、(g')前記コンパイルによって生成されたオブジェクトコードのコードサイズを比較する比較手段と、(h')この比較の結果、よりコードサイズの小さいオブジェクトコードに対応するソースプログラムの記述手法を所望のコーディング規則としてこれを適宜の記憶手段に格納するコーディング規則格納手段と、を具えるようにすることで、各種手続別の好ましいコーディング規則についてこれを全て自動的に作成することもできるようになる。

【0022】また、同コーディング支援装置の上記構成によれば、ユーザによっては知らない記述方法が上記同義異文として自動生成されることも有り得る。したがっ

て、上記(a)の入力手段、(b)の字句解析手段、(c)の構文解析手段、(d)の意味解析手段、及び(e)の同義異文生成手段に加えて、(j)適宜の出力操作に基づき、前記同義異文として生成されたソースプログラムを所定の形式にて出力する出力手段、を具える構成であっても、コーディング支援装置としての必要最小限の機能は満たされるようになる。こうして出力される同義異文は、特に経験の少ないユーザ（プログラマ）にとって、当該記述言語の言語仕様を理解する上での大きな助けとなる。

10 勿論、この構成に対して、上記(f)のコンパイル手段をはじめ、(g)の表示制御手段や(h)のコーディング規則格納手段、或いは(g')の比較手段や(h')のコーディング規則格納手段を付加した構成として同コーディング支援装置を実現することもできる。

【0023】

【実施例】図1に、この発明にかかるコーディング支援装置の一実施例を示す。この実施例の装置は、ユーザ（プログラマ）の所望するコーディング規則を、いわゆるコンピュータ支援によって作成する装置として構成されている。

20 【0024】すなわち同装置において、入力装置1は、例えばキーボードやマウス等からなって、ソースプログラムSPの入力などに用いられる周知の装置である。なお、この入力されるソースプログラムSPは、一旦入力されて図示しないディスク装置等に保管されているものであってもよい。その場合であっても、この入力装置1を用いての適宜の入力操作を通じて、ディスク装置等にあるソースプログラムSPが同実施例の装置に読み込まれるようになる。

30 【0025】また、表示装置2は、例えばCRTディスプレイや液晶ディスプレイ等からなって、上記入力されるソースプログラムSPの内容をユーザにエコーバックしたり、後述するオブジェクトコードのニーモニックや操作案内等を可視表示するこれも周知の出力装置である。

【0026】また、同じく出力装置である印字装置3も周知であり、この実施例の装置では特に、同装置を通じて作成されるコーディング規則の印字出力に用いられるものとする。

40 【0027】また、入出力制御部4は、これら入力装置1、表示装置2、及び印字装置3とコンピュータ本体とのインターフェースとして、

- ・入力装置1による入力内容や操作内容を解読し、該解読した内容をコンピュータ本体に取り込む。

- ・コンピュータ本体から表示指令のあった情報についてはこれを、表示装置2を通じて表示制御する。

- ・コンピュータ本体から印字指令のあった情報についてはこれを、印字装置2を通じて印字制御する。

といった機能を実現する部分である。

50 【0028】一方、同実施例の装置において、支援制御

部5、データベース6、字句解析部7、構文解析部8、意味解析部9、同義異文生成部10、及びコンパイル部11は何れも、コンピュータ本体にその制御プログラムを通じて仮想的に構成された部分である。

【0029】まず、支援制御部5は、上記入出力制御部4との間での各種情報の授受を行いつつ、以下に説明する各部を統括的に管理且つ制御して、上記入力されるソースプログラムSPに基づくコーディング規則の作成を行う部分である。なお、この支援制御部5は、その内部に作業領域を有し、この作業領域にソースプログラムSPを読み込んでこれを所要に内部処理する。

【0030】また、データベース6は、同図1に付記するように、対象となるプログラム言語についての単語辞書、構文規則、内部表現（論理構造や文情報等）、及びコーディング規則が格納されている部分である。以下に説明する各解析部では、これら格納されている情報をそ*

```
main() {
    int x, i; /* 整数型の変数 x 及び i を定義 */

    for(i=0; i<=10; ++i) {
        /* i を 0 に初期化し、これが 10 以下である間、
           i をインクリメント */
        x = x + i;
        /* 変数 x に (x + i) を代入 */
    }
}
```

このようなソースプログラムSPの記述に対し、この字句解析部7では、以下の態様で、これをトークンに分解する。

```
main
(
)
{
int
x
,
i
;
for
...

```

因みにC言語の場合、トークンとしては、「識別子」、「キーワード」、「定数」、「文字列リテラル」、「演算子」、及び「他の区切り子」の6種がある。

【0034】構文解析部8は、こうして分解されたトークンに基づいて、ソースプログラムSPの構文構造を明らかにする部分である。この構文の解析には、データベース6に格納されている構文規則が参照される。

【0035】例えばここでの例の場合、構文解析部8では、上記一連のトークンから、このソースプログラムSPの構文構造が、「for文からなるループ制御」であ

*の都度参照して、入力されたソースプログラムSPについてのそれぞれ該当する解析を実行するようになる。

【0031】字句解析部7、構文解析部8、及び意味解析部9の各解析部は、データベース6に格納されている上記情報に基づき、それぞれ次のような解析を実行する部分である。

【0032】字句解析部7は、上記作業領域（支援制御部5）に読み込まれているソースプログラムSPの字句を解析して、同ソースプログラムSPを当該記述言語としての一連のトークン（綴り）に分解する部分である。この字句の解析には、データベース6に格納されている単語辞書が参照される。

【0033】例えば、対象となるプログラム言語がC言語であり、ソースプログラムSPには、図2に例示するような記述があったとする。参考までにその意味を解説すると次のようである。

る旨を解析する。

【0036】そして、意味解析部9は、上記分解された一連のトークン、並びに上記明らかにされた構文構造に基づいて、当該ソースプログラムSPによる処理内容（手続）の全てを明らかにする部分である。この意味の解析には、データベース6に内部表現として格納されている論理構造や文情報等が参照される。

【0037】例えばここでの例の場合、意味解析部9では、上記記述からなるソースプログラムSPに対し、図3に例示する態様にて、その処理内容を明らかにする。この図3は、支援制御部5内の作業領域に実際には内部表現として展開される同解析結果を模式的に示したものである。同図3に示されるように、この段階で「処理項目」が「for文からなるループ制御」であること、また「具体的処理」として、

- ・「初期値の設定」は「i = 0;」
- ・「式の評価」は「i <= 10;」
- ・「付加実行文」は「++i;」
- ・「実行単位」は「x = x + i;」

であることが全て明らかにされる。

【0038】こうして明らかにされた当該ソースプログラムSPの処理内容は、上記支援制御部5を通じて同義異文生成部10に引き渡される。同義異文生成部10は、こうしたソースプログラムSPについての解析結果

に基づいて、当該記述言語による同義異文、すなわちソースプログラムSPと同じ処理内容を有して記述方法の異なるプログラム（ソースプログラム）を生成する部分である。

【0039】例えばここでの例の場合、同義異文生成部10では、上記解析結果（図3）に基づいて、まず「for文」以外のループ制御の手段としての「while文」をデータベース6から検索し、図4に示される態様にて、この「while文」による構文テンプレートを作成する。そしてその後、この作成したテンプレートに、同解析の結果明らかになった具体的処理内容、すなわち上記「初期値の設定（i=0;）」、「式の評価（i<=10;）」、「付加実行文（++i;）」、及び「実行単位（x=x+i;）」をそれぞれ埋め込んで、図5に示されるような同義異文を完成する。この図5に示されるプログラムの処理内容が、先の図2に示したソースプログラムSPの処理内容と意味的に同一のものであることはいうまでもない。

【0040】こうして同義異文として生成されたプログラムは次に、先の入力されたソースプログラムSPと共に、コンパイル部11に渡される。コンパイル部11は、これら2種類のプログラムを各別にコンパイルしてそれぞれのオブジェクトコードを生成する部分である。なおこのコンパイルは、双方のプログラムに対し同一の条件で実行される。すなわちソースプログラムSPの入 *

・オブジェクトコード1

```

1:      LD   (i),10   ; (i) に10をロード
2: LABEL1 JZE  LABEL2 ; (i) = 0であればLABEL2にジャンプ
3:      ADD  (x),(i)   ; (x) に (i) 加える
4:      SUB  (i),1     ; (i) から1を減ずる
5:      JMP  LABEL1    ; LABEL1にジャンプ
6: LABEL2                ; 終了

```

・オブジェクトコード2

```

1:      LD   (i),0     ; (i) に0をロード
2:      JMP  LABEL2    ; LABEL2にジャンプ
3: LABEL1 ADD  (x),(i)   ; (x) に (i) 加える
4:      ADD  (i),1     ; (i) に1を加える
5: LABEL2 CMP  (i),10   ; (i) と10とを比較する
6:      JLE  LABEL1    ; 10以下であればLABEL1にジャンプ

```

また、同図6に併せ示すように、この表示画面には、これら「オブジェクトコード1」及び「オブジェクトコード2」の何れがサイズの効率的なよいコードであるかをユーザに対して選択要求する表示が含まれる。因みに上記の例の場合には、「オブジェクトコード1」の方がより効率的なよいコードとして生成されており、ユーザとしても、この選択要求に対しては「1」を選択することとなる。該選択操作は、入力装置1（キーボードやマウス等）を通じて行われる。

*力に際して何らかのコンパイルオプションが指定されていれば、上記同義異文として生成されたプログラムに対しても同一のコンパイルオプションが指定されるようになっている。コンパイルオプションとは周知のように、「コードサイズの優先」とか「実行速度の優先」とかをコンパイル部11に対して指定するための所定の情報である。

【0041】そして、このコンパイルによって生成されたオブジェクトコードは、支援制御部5によって、いわゆるアセンブラ展開形に変換（逆アセンブル）され、その変換結果が、例えば図6に示される態様にて、表示装置2に表示制御される。

【0042】すなわち図6は、表示装置2の表示画面の一例として、入力されたソースプログラムSPを「ソースプログラム1」とし、生成された同義異文を「ソースプログラム2」として、それらの各コンパイル結果を対比表示している様子を模式的に示したものである。このように、コンパイル結果である各オブジェクトコードについては、アセンブラ展開形としてのニーモニック表現を用いて表示することで、オブジェクトコードとして見た場合のそれら両プログラムの違い、優劣等はユーザにとっても瞭然となる。参考までに、図6に例示した「オブジェクトコード1」及び「オブジェクトコード2」の内容を解説するとそれぞれ以下のようなものである。

【0043】こうして、ユーザによる選択操作がある、その旨を入出力制御部4を介して認知した支援制御部5は、その選択されたオブジェクトコードに対応するソースプログラムの記述方法を、採用すべきコーディング規則として、データベース6に格納する。因みに上記の例の場合には、

・「ループ制御」として「for文」と「while文」とを比較した結果、「for文」を採用すべきコーディング規則とする。旨の情報が、データベース6に格

納されることとなる。

【0044】この実施例のコーディング支援装置では、その都度入力される各種ソースプログラムSPに対し、全て上記に準じたかたちで、その解析、同義異文生成、コンパイル、並びにユーザ選択に基づくコーディング規則の格納処理を繰り返すようになる。

【0045】そしてその後、ユーザによるコーディング規則の出力要求があれば、これを受けた支援制御部5では、データベース6に格納されているコーディング規則が、例えば図7に示されるような一覧表形式にて出力されるよう、その必要とされる情報を入出力制御部4に渡す。該出力要求が印字要求であれば、印字装置3を通じたプリントアウトPRTとしてこのコーディング規則が出力され、同出力要求が表示要求であれば、表示装置2を通じた画面出力として同コーディング規則が出力されるようになる。

【0046】図8は、同実施例の装置によるコーディング支援手順（コーディング規則作成手順）を総括して示したものであり、以下、この図8を併せ参照して、装置全体としての動作、並びにその処理手順を更に説明する。

【0047】すなわちいま、入力装置1を通じてソースプログラムSPの入力があつた旨を判断すると（ステップ100）、支援制御部5は、この入力されたソースプログラムSPを前記作業領域に読み込み、字句解析部7を起動して、その字句解析を開始する（ステップ101）。こうした字句解析によって、同ソースプログラムSPの記述が当該記述言語としての一連のトークンに分解されるようになることは上述した通りである。

【0048】こうして、当該ソースプログラムSPについての字句解析を終えた支援制御部5は次に、構文解析部8を起動し、該構文解析部8を通じて、上記分解されたトークンに基づく構文解析を実行する（ステップ102）。この構文解析の実行によって、当該ソースプログラムSPの構文構造が明らかになる。例えば図2に例示したソースプログラムSPにおいて、そこに「for文によるループ制御」が記述されていることもこの段階で明らかになる。

【0049】こうして構文解析を終えると、支援制御部5は更に、意味解析部9を起動し、該意味解析部9を通じて、上記分解された一連のトークン、並びに上記明らかにされた構文構造に基づく意味解析を実行する（ステップ103）。この意味解析の実行によって、当該ソースプログラムSPの処理内容の全てが明らかになる。例えば図2に例示したソースプログラムSPに対しては、図3に示される態様でその処理内容の全てが評価されるようになることも上述した。

【0050】こうして、当該ソースプログラムSPについての全ての解析を終えると、支援制御部5は次に、同義異文生成部10を起動し、該同義異文生成部10を通

じてソースプログラムSPの同義異文を生成する（ステップ104）。これについても既に詳述したように、例えば図2に例示されるソースプログラムSPに対しては、その解析結果に基づき、例えば図5に示される態様で、その同義異文が生成されるようになる。

【0051】こうして同義異文を得た支援制御部5は更に、コンパイル部11を起動し、該コンパイル部11を通じて、この同義異文とソースプログラムSPとを各々コンパイルする（ステップ105）。なお、ソースプログラムSPの入力の際、これに何らかのコンパイルオプションが指定されていた場合には、同義異文に対しても同一のコンパイルオプションを指定してこれをコンパイルする。

【0052】コンパイルの結果、オブジェクトコードが生成されると、支援制御部5はこれを受け取り、それらオブジェクトコードをアセンブラ展開形に変換（逆アセンブル）した後、図6に例示した態様で、それらニモニック表現したオブジェクトコードを表示装置2に対比表示する（ステップ106）。なお、この表示画面には、ユーザに対するそれらオブジェクトコードの選択要求も併せ表示される。

【0053】この選択要求に対して、ユーザによる選択操作があれば（ステップ107）、その旨判断した支援制御部5は、その選択されたオブジェクトコードに対応するソースプログラムの記述方法を、採用すべきコーディング規則として、データベース6に格納する（ステップ108）。

【0054】そしてその後、ユーザによる終了操作があれば（ステップ109）、当該支援処理を終了し、ユーザによる印字要求を示す操作があれば（ステップ110）、図7に例示したようなコーディング規則を印字装置3を通じて印字出力した後（ステップ111）、当該支援処理を終了する。なお、この図8では割愛したが、ユーザによる表示要求を示す操作があつた場合には、同コーディング規則は表示装置2を通じて画面出力されるようになる。

【0055】以上のように、この実施例のコーディング支援装置によれば、ユーザによって記述されたソースプログラムの他に同義異文が自動生成されるようになる。また、同実施例のコーディング支援装置によれば、これら元のソースプログラムと生成された同義異文とは同一条件にてコンパイルされた後、その各オブジェクトコードがニモニックとして対比表示されるようになる。

【0056】このため、オブジェクトコードとして見た場合のそれら両プログラムの違い、優劣等はユーザにとっても瞭然となり、ソースプログラムとして何れの記述方法が効率のよいオブジェクトコードを生成し得る記述方法であるかが容易に判断できるようになる。

【0057】そして、これらオブジェクトコードについての選択操作が行われ、上記データベースに格納される

ソースプログラムの記述手法とは自ずと、ユーザが要求する処理内容についてのより好ましいコーディング規則を示すものとなる。

【0058】また、この実施例のコーディング支援装置によれば、こうしてデータベースに蓄積されるコーディング規則についての情報が、ユーザによる出力要求操作に応じて一覧表として出力されるようになる。

【0059】したがって、各ユーザが全て、以後、この出力されたコーディング規則をもとにソースプログラムを記述するようにすれば、可読性の悪化やオブジェクトコードのばらつき、更にはそれら記述方法の修正等によって引き起こされる生産能率の低下などといった問題も自ずと解消されるようになる。

【0060】ところで、上記実施例の装置では、サイズのより効率のよいオブジェクトコードの選択をユーザに委ねる構成をとっているが、こうしたコードサイズの比較であれば、上記支援制御装置 5 自身がその大小を判断することも可能である。

【0061】そこで次に、この発明にかかるコーディング支援装置の第 2 の実施例として、支援制御装置 5 自身が各オブジェクトコードのコードサイズを比較して、サイズのより効率のよいオブジェクトコードを選択する装置について説明する。

【0062】ただし、この第 2 の実施例の装置において、その基本的な構成は、図 1 に示した先の実施例の構成に準ずるものであり、ここでの改めでの図示、並びにそれら構成についての重複する説明は割愛する。

【0063】図 9 は、この第 2 の実施例の装置によるコーディング支援手順（コーディング規則作成手順）について示したものであり、以下では、この図 9 に基づいて、同第 2 の実施例の装置の動作、並びにその処理手順を説明する。ただしこの図 9 において、そのステップ 200 ～ステップ 205 にかかる処理は、先の実施例の装置の図 8 におけるステップ 100 ～ステップ 105 にかかる処理と同一であり、これらの各処理についてもその重複する説明は割愛する。

【0064】さて、この第 2 の実施例の装置にあって、支援制御部 5 は、コンパイルの実行（ステップ 205）によって得られた各オブジェクトコードのコードサイズを自ら比較する（ステップ 206）。いわゆる複数ファイルのファイルサイズを比較する手法自体は周知であり、支援制御部 5 によるこうした比較も、それら周知の手法を用いて容易に実現することができる。

【0065】こうしてコードサイズの比較を実行した支援制御部 5 は引き続き、よりコードサイズの小さいオブジェクトコードに対応するソースプログラムの記述方法を選択し（ステップ 207）、該選択したソースプログラムの記述方法を、採用すべきコーディング規則としてデータベース 6 に格納する（ステップ 208）。

【0066】そしてその後は、先の実施例の装置の場合

と同様、ユーザによる終了操作があれば（ステップ 209）、当該支援処理を終了する。また、ユーザによる印字要求を示す操作があれば（ステップ 210）、図 7 に例示したようなコーディング規則を印字装置 3 を通じて印字出力した後（ステップ 211）、当該支援処理を終了する。

【0067】このように、この第 2 の実施例の装置によれば、効率的なオブジェクトコードを生成し得るコーディング規則について、これを全て自動的に作成することができるようになる。

【0068】また、これら各実施例の装置の上記構成によれば、ユーザによっては知らない記述方法が前述した同義異文として自動生成されることも有り得る。したがって、同装置のこうした機能を積極的に利用すれば、すなわちユーザによる出力要求に応じてそれら同義異文も可視出力されるようにすれば、特に経験の少ないユーザにとっては、その出力される同義異文が当該記述言語の言語仕様を理解する上での大きな助けとなる。

【0069】そこで更に、この発明にかかるコーディング支援装置の第 3 の実施例として、同義異文が生成された時点でユーザによる出力要求があれば、その生成された同義異文を、例えばユーザが入力したソースプログラムと対比して出力することのできる装置について説明する。

【0070】ただし、この第 3 の実施例の装置においても、その基本的な構成は、図 1 に示した実施例の構成に準ずるものであり、ここでの改めでの図示、並びにそれら構成についての重複する説明は割愛する。

【0071】図 10 は、この第 3 の実施例の装置によるコーディング支援手順について示したものであり、以下では、この図 10 に基づいて、同第 3 の実施例の装置の動作、並びにその処理手順を説明する。なお、この図 10 においても、そのステップ 300 ～ステップ 304 にかかる処理は、先の実施例の装置の図 8 におけるステップ 100 ～ステップ 104 にかかる処理と同一である。したがって、これらの各処理についてもその重複する説明は割愛する。

【0072】さて、この第 3 の実施例の装置にあって、支援制御部 5 は、同義異文が自動生成された時点（ステップ 304）でユーザによる出力要求（出力指令）の有無を確認する（ステップ 305）。

【0073】そして、出力要求があった旨判断した支援制御部 5 は、該生成された同義異文が、例えばユーザが入力したソースプログラム SP と対比されるよう、表示装置 2 或いは印字装置 3 を通じてこれを出力する（ステップ 306）。

【0074】支援制御部 5 はその後、ユーザによる終了操作があれば（ステップ 307）、当該支援処理を終了し、同終了操作がなかった場合、或いは上記出力要求がなかった場合は、先の実施例と同様、それらプログラム

(ソースプログラム及びその同義異文) についてのコンパイルを実行する(図8ステップ105、または図9ステップ205)。

【0075】このように、この第3の実施例によれば、自動生成された同義異文がユーザによる出力要求に応じて可視出力されるようになる。なお、特にこの第3の実施例にあつては、コンパイル部11を削除し、図10のステップ307或いはステップ305までの処理に対応した機能しか具えない小規模な装置として同コーディング支援装置を構成することもできる。このような構成であっても、コーディング支援装置としてのこうした必要最小限の機能は満たされ、しかもこうした構成によれば、例えばBASIC(ベーシック)等のインタプリタ型のプログラム言語についても、同コーディング支援装置の適用が可能となる。

【0076】ところで、上記何れの実施例であれ、同義異文生成部10を通じて自動生成される同義異文の数は1文だけとは限らない。この同義異文は、ソースプログラムSPの構文構造等に応じてその同義の構文が検索されるものであり、同義の構文が更に複数存在する場合には、この同義異文としても自ずと複数の文が生成されるようになる。

【0077】また、データベース6に格納されたコーディング規則については、必ずしもこれを表示装置2や印字装置3を通じて出力する必要はない。例えば、こうしてデータベース6に格納された状態であっても、従来のコーディング規則検査装置において記憶手段に登録されるコーディング規則ファイルとして用いることはできるようになる。

【0078】また、図7に例示したコーディング規則の出力例も一例にすぎない。例えば、「多方向分岐」の項目についていえば、

・3方向迄の分岐であれば、採用記述方法として「if-e l s e文」を用い、4方向以上の分岐となるときに、同採用記述方法として「s w i t c h文」を用いる。などといった態様で、更に木目細かい規則としてこれを作成し、または出力するようにしてもよい。

【0079】

【発明の効果】以上説明したように、この発明によれば、より効率のよいオブジェクトコードを生成する上で望ましいコーディング規則を容易に得ることができ、ユーザ(プログラマ)によるその後のソースプログラムの記述をより強力に支援することができるようになる。

【0080】またこの発明によれば、ユーザによって記述されたソースプログラムと実質的に同一の手続が記述された同義異文が自動生成されるため、経験の少ないユーザも安心して利用することができるようになる。特に経験の少ないユーザにとって、こうして自動生成される同義異文は、当該記述言語の言語仕様を理解する上での大きな助けとなる。

【図面の簡単な説明】

【図1】この発明にかかるコーディング支援装置についてその一実施例構成を示すブロック図である。

【図2】同実施例のコーディング支援装置に入力されるソースプログラムの一例を模式的に示す略図である。

【図3】図2に例示したソースプログラムについてその解析結果の一例を模式的に示す略図である。

【図4】図3に例示した解析結果に基づいて生成される同義異文の構文構造についてその一例を模式的に示す略図である。

【図5】図4に例示した構文構造に図3に例示した解析結果を埋め込んで生成される図2に例示したソースプログラムの同義異文の一例を模式的に示す略図である。

【図6】図1に示される表示装置の表示画面の一例として、入力されたソースプログラムと生成された同義異文との各コンパイル結果についてのニモニック表現、並びにユーザに対する選択要求の画面表示例を示す略図である。

【図7】図1に示される印字装置による出力例として、同実施例のコーディング支援装置によって作成されたコーディング規則の一例を示す略図である。

【図8】同実施例のコーディング支援装置によるコーディング支援手順(コーディング規則作成手順)についてその一例を示すフローチャートである。

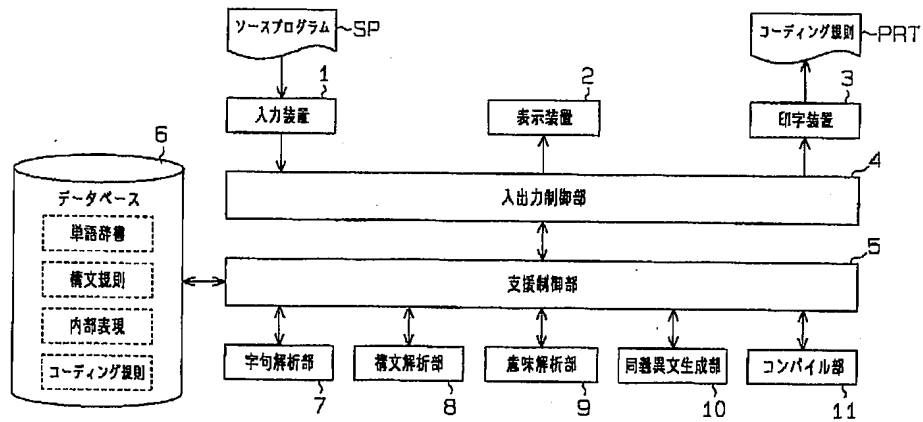
【図9】この発明にかかるコーディング支援装置の第2の実施例によるコーディング支援手順(コーディング規則作成手順)を示すフローチャートである。

【図10】この発明にかかるコーディング支援装置の第3の実施例によるコーディング支援手順を示すフローチャートである。

【符号の説明】

1…入力装置、2…表示装置、3…印字装置、4…入力制御部、5…支援制御部、6…データベース、7…字句解析部、8…構文解析部、9…意味解析部、10…同義異文生成部、11…コンパイル部。

【図1】



【図2】

```

main() {
    int x, i;
    for (i=0; i<=10; ++i) {
        x = x + i;
    }
}

```

~SP

【図3】

for	: ループ制御	処理項目
i=0;	: 初期値の設定	} 具体的処理
i<=10;	: 式の評価	
++i;	: 付加実行文	
x = x + i;	: 実行単位	

【図4】

```

main() {
    int x, i;
    {
        初期値の設定
        while (式の評価) {
            実行単位
            付加実行文
        }
    }
}

```

【図5】

```

main() {
    int x, i;
    i=0;
    while (i<=10) {
        x = x + i;
        ++i;
    }
}

```

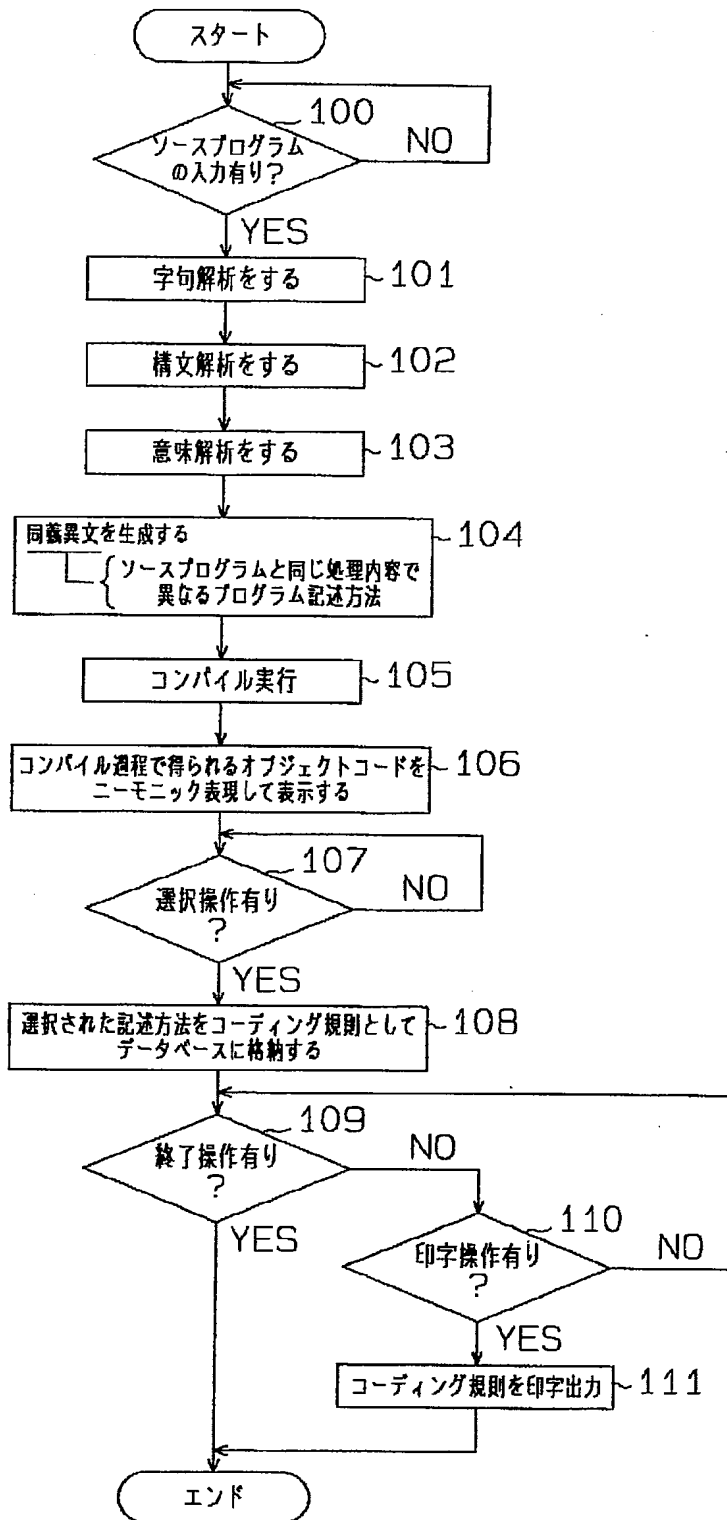
【図6】

ソースプログラム1 <pre>for (i=0; i<=10; ++i){ x = x + i; }</pre>	ソースプログラム2 <pre>i=0; while (i<=10){ x = x + i; ++i; }</pre>	<div style="text-align: center;"> $\begin{matrix} 2 \\ \downarrow \end{matrix}$ </div> <p>いずれかの オブジェクトコード (ソースプログラム) の番号を選択 して下さい。</p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px 5px;">1</div> <div style="border: 1px solid black; padding: 2px 5px;">2</div> </div>	
オブジェクトコード1 <pre>1: LD (i), 10 2: LABEL1 JZE LABEL2 3: ADD (x), (i) 4: SUB (i), 1 5: JMP LABEL1 6: LABEL2</pre>	オブジェクトコード2 <pre>1: LD (i), 0 2: JMP LABEL2 3: LABEL1 ADD (x), (i) 4: ADD (i), 1 5: LABEL2 CMP (i), 10 6: JLE LABEL1</pre>		

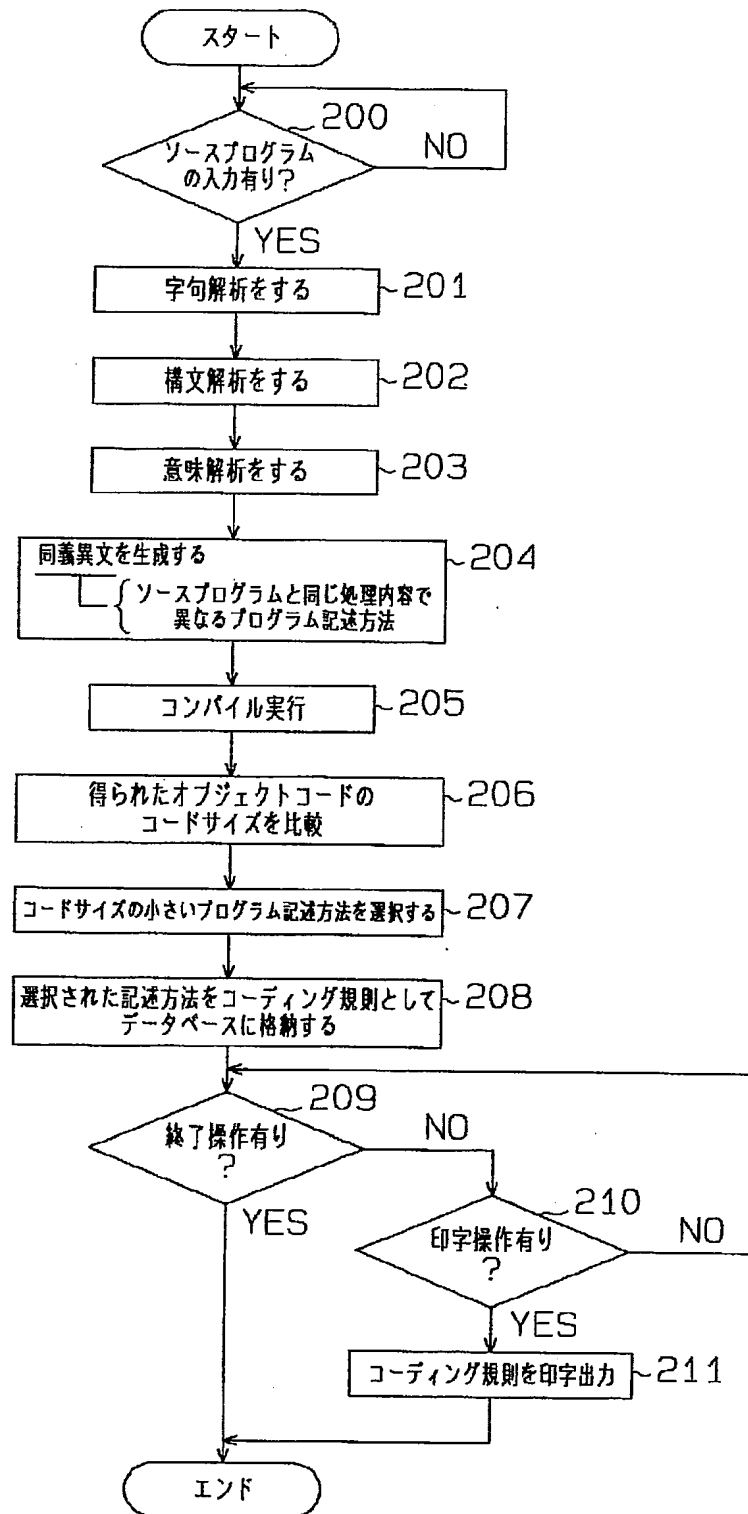
【図7】

コーディング規則		
項目	採用記述方法	比較記述方法
...
メンバー参照	直接メンバー演算子	間接メンバー演算子
多方向分岐	switch文	if-else文
ループ制御	for文	while文
...

【図 8】



【図 9】



【図10】

